

TRƯỜNG ĐẠI HỌC VIỆT NHẬT - ĐHQGHN  
CHƯƠNG TRÌNH ĐÀO TẠO  
KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

---



## **BÁO CÁO DỰ ÁN CUỐI KỲ** **Hệ thống Quản lý Nhà sách**

**Môn học:** Lập trình nâng cao

**Mã lớp học phần:** CSE3010.2

**Giảng viên:** Chu Thị Minh Huệ

**Sinh viên:** Nguyễn Quang Anh

**Mã số sinh viên:** 23110149

**Hà Nội, 2026**

## LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn chân thành đến Ban Giám hiệu và quý thầy cô Trường Đại học Việt Nhật - Đại học Quốc gia Hà Nội đã luôn tận tình giảng dạy, tạo môi trường học tập tốt nhất và trang bị cho em những nền tảng kiến thức quý báu.

Đặc biệt, em xin bày tỏ lòng biết ơn sâu sắc nhất tới giảng viên Chu Thị Minh Huệ. Trong suốt quá trình học tập môn Lập trình Nâng cao và thực hiện đề tài này, cô đã luôn tận tâm hướng dẫn, định hướng giải pháp và đưa ra những góp ý chuyên môn thiết thực. Cô đã luôn tận tình hỗ trợ, định hướng và góp ý chuyên môn, đồng thời cũng tạo điều kiện tốt nhất để em có thể tìm hiểu và thực hiện báo cáo này trong suốt thời gian thực hiện dự án

Mặc dù em đã nỗ lực hoàn thành báo cáo với tinh thần trách nhiệm cao nhất, nhưng do hạn chế về kinh nghiệm thực tế, sản phẩm khó tránh khỏi những thiếu sót. Em rất mong nhận được những ý kiến đóng góp cô để có thể rút kinh nghiệm và phát triển hoàn thiện hơn trong các dự án tương lai.

Em xin chân thành cảm ơn!

## **LỜI CAM ĐOAN**

Em xin cam đoan rằng toàn bộ nội dung được trình bày trong báo cáo này là kết quả học tập và làm việc của cá nhân em, dưới sự hướng dẫn của giảng viên Chu Thị Minh Huệ.

Các số liệu và kết quả trình bày trong báo cáo là trung thực. Em xin chịu hoàn toàn trách nhiệm trước Giảng viên và Nhà trường về tính trung thực và độ chính xác của các nội dung trong báo cáo này.

Hà Nội, ngày 10 tháng 1 năm 2026

## TÓM TẮT

Trong bối cảnh chuyển đổi số hiện nay, việc quản lý thủ công tại các cửa hàng sách truyền thống thông qua sổ sách hoặc các công cụ văn phòng đơn giản (như Excel) đang bộc lộ nhiều hạn chế. Các vấn đề thường gặp bao gồm việc kiểm soát tồn kho thiếu chính xác, quy trình thanh toán mất nhiều thời gian, khó khăn trong việc tra cứu thông tin sách và sai sót trong tổng hợp báo cáo doanh thu. Những bất cập này đặt ra yêu cầu cấp thiết về một giải pháp phần mềm chuyên nghiệp giúp tự động hóa quy trình nghiệp vụ, nâng cao hiệu quả quản lý và tối ưu hóa trải nghiệm bán hàng. Xuất phát từ thực tiễn đó, em đã thực hiện đề tài xây dựng phần mềm **Quản lý Nhà sách (BookStore Management)** trên nền tảng Windows.

Về mặt kỹ thuật, hệ thống được phát triển sử dụng ngôn ngữ C# và công nghệ Windows Forms, tuân thủ chặt chẽ mô hình kiến trúc 3 lớp. Cấu trúc dự án được tổ chức và tách biệt thành các Components độc lập: Presentation Layer (UI), Business Logic Layer (BLL), Data Access Layer (DAL) và Entities. Về phương diện kỹ thuật, cơ chế truy xuất dữ liệu của hệ thống được xây dựng trên nền tảng **Entity Framework Core** theo phương pháp **Code First**, kết hợp chặt chẽ với kỹ thuật **Dependency Injection (DI)** nhằm gia tăng tính linh hoạt, khả năng mở rộng và dễ dàng bảo trì mã nguồn. Điểm nhấn công nghệ của hệ thống là việc ứng dụng triệt để mô hình lập trình bất đồng bộ cho các tác vụ xử lý dữ liệu và trích xuất báo cáo. Giải pháp này đảm bảo giao diện người dùng luôn hoạt động mượt mà, khắc phục hoàn toàn hiện tượng "treo" ứng dụng khi phải xử lý khối lượng tính toán lớn hoặc thao tác phức tạp.

Về khía cạnh chức năng, phần mềm cung cấp một giải pháp quản lý toàn diện, bao quát từ việc quản lý các danh mục dữ liệu nền tảng (đầu sách, tác giả, nhà xuất bản, khách hàng) đến các quy trình nghiệp vụ cốt lõi như lập hóa đơn bán hàng và kiểm soát biến động tồn kho theo thời gian thực. Bên cạnh đó, hệ thống được tích hợp chức năng báo cáo thống kê trực quan, hỗ trợ kết xuất dữ liệu doanh thu sang các định dạng lưu trữ phổ biến như Excel và PDF.

# MỤC LỤC

<b>CHƯƠNG I. YÊU CẦU.....</b>	<b>6</b>
<b>1. Yêu cầu người dùng.....</b>	<b>6</b>
a. Yêu cầu về Quản lý hệ thống và Bảo mật thông tin.....	6
b. Yêu cầu về Quản lý Danh mục hàng hoá.....	7
c. Yêu cầu về Nghiệp vụ Bán hàng và Giao dịch.....	7
d. Yêu cầu về Quản lý Khách hàng.....	7
e. Yêu cầu về Báo cáo và Thống kê.....	7
f. Yêu cầu về Hiệu năng và Trải nghiệm người dùng.....	8
<b>2. Phân tích các chức năng.....</b>	<b>8</b>
a. Nhóm chức năng Quản trị hệ thống và Phân quyền.....	8
b. Nhóm chức năng Quản lý Danh mục.....	9
c. Nhóm chức năng Nghiệp vụ Bán hàng.....	9
d. Nhóm chức năng Báo cáo và Thống kê.....	9
<b>CHƯƠNG II. MÔ HÌNH THỰC THỂ.....</b>	<b>10</b>
<b>1. Tổng quan cơ sở dữ liệu.....</b>	<b>10</b>
<b>2. Danh sách các thực thể (Entities).....</b>	<b>10</b>
a. Nhóm Quản trị hệ thống & Phân quyền.....	10
b. Nhóm Quản lý Danh mục Sách.....	10
c. Nhóm Nghiệp vụ Bán hàng.....	11
d. Các lớp hỗ trợ thống kê DTO (Data Transfer Objects).....	11
<b>3. Cài đặt chi tiết (Implementation).....</b>	<b>11</b>
a. Cấu trúc lớp thực thể (Entity Class).....	12
b. Cấu hình quan hệ và ràng buộc (Fluent API).....	12
<b>CHƯƠNG III. THIẾT KẾ CÁC THÀNH PHẦN PHẦN MỀM.....</b>	<b>13</b>
<b>1. Kiến trúc tổng thể (Architecture).....</b>	<b>13</b>
<b>2. Tổ chức Dự án (Solution Structure).....</b>	<b>14</b>
<b>3. Mô tả chi tiết quy trình xử lý (Minh hoạ chức năng).....</b>	<b>14</b>
Bước 1: Tầng Entities & DAL (Cơ sở dữ liệu).....	15
Bước 2: Tầng BLL (Xử lý nghiệp vụ & Transaction).....	16
Bước 3: Tầng UI (Giao diện người dùng).....	17
<b>4. Các kỹ thuật lập trình nâng cao.....</b>	<b>18</b>
a. Dependency Injection (DI).....	18
b. Kỹ thuật Database Transaction.....	19
c. Truy vấn dữ liệu với LINQ (Language Integrated Query).....	19
d. Mô hình bất đồng bộ (Asynchronous Programming).....	20
<b>CHƯƠNG IV. CHƯƠNG TRÌNH PHẦN MỀM.....</b>	<b>20</b>
<b>1. Màn hình Đăng nhập (Login).....</b>	<b>20</b>
<b>2. Màn hình Chính (Dashboard).....</b>	<b>21</b>
<b>3. Phân hệ Quản lý Danh mục (Admin &amp; Staff).....</b>	<b>22</b>
a. Quản lý Sách (Book Management).....	22
b. Quản lý các Danh mục bổ trợ.....	23
c. Quản lý Người dùng (User Management - Admin Only).....	25

<b>4. Phân hệ Bán hàng &amp; Thống kê.....</b>	<b>26</b>
a. Quản lý Khách hàng (Customer Management).....	26
b. Lập Hóa đơn bán hàng (Create Invoice).....	27
c. Báo cáo Doanh thu (Revenue Report).....	28
<b>CHƯƠNG V. TỔNG KẾT.....</b>	<b>29</b>
<b>1. Kết luận.....</b>	<b>29</b>
<b>2. Các hạn chế.....</b>	<b>30</b>
<b>3. Hướng phát triển.....</b>	<b>30</b>

# CHƯƠNG I. YÊU CẦU

## 1. Yêu cầu người dùng

Trong bối cảnh hoạt động kinh doanh thực tế tại các nhà sách hiện nay, việc quản lý thủ công bằng sổ sách hoặc các công cụ văn phòng đơn giản (như Excel) đang bộc lộ nhiều hạn chế về hiệu suất và độ chính xác. Xuất phát từ nhu cầu cấp thiết về việc chuẩn hóa quy trình vận hành và nâng cao hiệu quả quản lý, người dùng (bao gồm Chủ cửa hàng và Đội ngũ nhân viên bán hàng) đặt ra các yêu cầu nghiệp vụ cụ thể đối với "Phần mềm Quản lý Nhà sách" như sau:

### a. Yêu cầu về Quản lý hệ thống và Bảo mật thông tin

**Cơ chế xác thực người dùng:** Người dùng mong muốn hệ thống phải có quy trình đăng nhập bảo mật để đảm bảo chỉ những nhân viên được cấp phép mới có quyền truy cập vào dữ liệu nhạy cảm của cửa hàng.

**Phân cấp quyền hạn:** Cần có sự phân chia vai trò rõ ràng giữa "Quản trị viên" (Admin) và "Nhân viên bán hàng" (Staff). Cụ thể:

- **Quản trị viên:** Có toàn quyền kiểm soát hệ thống, bao gồm việc quản lý tài khoản nhân viên (thêm mới, khóa tài khoản, đặt lại mật khẩu) để kiểm soát nhân sự.
- **Nhân viên:** Chỉ được phép truy cập vào các chức năng phục vụ bán hàng và tra cứu thông tin, không được phép can thiệp vào các cấu hình hệ thống hoặc dữ liệu doanh thu nhạy cảm nếu không được cho phép.

### b. Yêu cầu về Quản lý Danh mục hàng hoá

**Số hóa dữ liệu sách:** Người dùng cần một cơ sở dữ liệu tập trung để lưu trữ thông tin chi tiết của đầu sách. Các thông tin cần quản lý bao gồm: Tên sách, Giá bán, Số lượng tồn kho, cùng các thông tin phân loại như Thể loại (Category), Tác giả (Author) và Nhà xuất bản (Publisher).

**Cập nhật dữ liệu linh hoạt:** Hệ thống cần cung cấp các công cụ (CRUD) cho phép người quản lý dễ dàng thêm mới sách khi nhập hàng, cập nhật giá bán hoặc thông tin sách khi có thay đổi, và xóa bỏ các mã sách không còn kinh doanh (với điều kiện ràng buộc toàn vẹn dữ liệu chặt chẽ).

**Tra cứu thông minh:** Nhân viên bán hàng cần khả năng tìm kiếm sách nhanh chóng theo Tên sách hoặc Tên tác giả để trả lời khách hàng ngay lập tức mà không mất thời gian tra sổ.

### c. Yêu cầu về Nghiệp vụ Bán hàng và Giao dịch

**Lập hóa đơn tự động:** Người dùng mong muốn quy trình thanh toán diễn ra nhanh gọn trên giao diện phần mềm: từ việc chọn khách hàng, chọn sách, nhập số lượng, đến việc hệ thống tự động tính toán tổng tiền (Thành tiền = Đơn giá x Số lượng).

**Kiểm soát tồn kho theo thời gian thực:** Hệ thống bắt buộc phải tự động kiểm tra số lượng tồn kho khả dụng trước khi bán. Nếu số lượng khách mua lớn hơn số lượng tồn, phần mềm cần đưa ra cảnh báo để tránh tình trạng bán không.

**Xử lý giao dịch an toàn:** Khi có hóa đơn, hệ thống phải đảm bảo tính nhất quán của dữ liệu: vừa ghi nhận doanh thu, vừa tự động trừ số lượng tồn kho tương ứng của các mặt hàng đã bán.

**In ấn hóa đơn:** Hỗ trợ xuất và in hóa đơn bán lẻ trực tiếp từ phần mềm để gửi cho khách hàng.

### d. Yêu cầu về Quản lý Khách hàng

Để phục vụ chiến lược chăm sóc khách hàng lâu dài, người dùng cần hệ thống lưu trữ thông tin cá nhân của khách hàng (Họ tên, Số điện thoại, Địa chỉ).

### e. Yêu cầu về Báo cáo và Thống kê

**Minh bạch hóa doanh thu:** Chủ cửa hàng cần các báo cáo thống kê chính xác về doanh thu bán hàng theo khoảng thời gian tùy chọn (theo ngày, tháng, năm) để nắm bắt tình hình kinh doanh.

**Phân tích xu hướng:** Cần có báo cáo thống kê "Top sách bán chạy" hoặc doanh thu theo thể loại sách, giúp người quản lý nhận định được thị hiếu khách hàng để có kế hoạch nhập hàng phù hợp.

**Lưu trữ và chia sẻ báo cáo:** Hệ thống cần hỗ trợ xuất các báo cáo thống kê ra các định dạng văn bản phổ biến như Excel (.xlsx) để tính toán thêm hoặc PDF (.pdf) để lưu trữ và in ấn báo cáo gửi lên cấp trên.

### f. Yêu cầu về Hiệu năng và Trải nghiệm người dùng

**Giao diện thân thiện:** Phần mềm cần có giao diện trực quan, dễ sử dụng, giúp nhân viên mới có thể làm quen và thao tác thành thạo trong thời gian ngắn.

**Hoạt động mượt mà:** Người dùng yêu cầu hệ thống phải phản hồi nhanh, không bị "treo" hay "đơ" máy ngay cả khi đang xử lý các tác vụ nặng như tải danh sách hàng nghìn cuốn sách hay xuất báo cáo doanh thu phức tạp.



## 2. Phân tích các chức năng

Hệ thống Quản lý Nhà sách được xây dựng với đầy đủ các phân hệ chức năng cần thiết để vận hành quy trình kinh doanh, từ quản lý danh mục, bán hàng đến báo cáo thống kê. Các chức năng được phân tích chi tiết như sau:

### a. Nhóm chức năng Quản trị hệ thống và Phân quyền

Là nhóm chức năng nền tảng đảm bảo tính bảo mật và kiểm soát truy cập của hệ thống.

**Đăng nhập (Login):** Hệ thống yêu cầu xác thực người dùng thông qua Tên đăng nhập và Mật khẩu trước khi truy cập vào giao diện chính.

**Phân quyền người dùng:** Hệ thống tự động nhận diện vai trò của người dùng sau khi đăng nhập để cấp quyền tương ứng

- Quản trị viên (Admin): Có toàn quyền truy cập tất cả các chức năng, bao gồm Quản lý người dùng, Xem báo cáo doanh thu và thực hiện các thao tác Thêm/Sửa/Xóa dữ liệu danh mục.
- Nhân viên (Staff): Chỉ được phép truy cập các chức năng bán hàng (Lập hóa đơn), Quản lý khách hàng và Tra cứu thông tin sách. Các chức năng nhạy cảm như Báo cáo doanh thu và Quản lý tài khoản hệ thống sẽ bị ẩn và vô hiệu hóa.

**Quản lý người dùng (Dành cho Admin):**

- Cho phép thêm mới tài khoản nhân viên, phân vai trò (Role).
- Hỗ trợ đặt lại mật khẩu (Reset Password) cho nhân viên trong trường hợp quên mật khẩu.
- Xóa tài khoản người dùng khỏi hệ thống.

### b. Nhóm chức năng Quản lý Danh mục

Nhóm chức năng này cung cấp các công cụ để quản lý dữ liệu nền tảng của nhà sách. Các thao tác CRUD (Create - Read - Update - Delete) được áp dụng triệt để cho các đối tượng:

**Quản lý Sách:**

- Thêm mới: Nhập thông tin chi tiết sách (Tên, Giá, Số lượng, Tác giả, Nhà xuất bản, Thể loại).
- Cập nhật: Cho phép sửa đổi thông tin sách (ví dụ: cập nhật giá bán, bổ sung số lượng tồn kho).
- Xóa: Cho phép xóa sách khỏi danh mục, có kiểm tra ràng buộc toàn vẹn dữ liệu (không cho phép xóa sách đã từng phát sinh giao dịch bán hàng).
- Tìm kiếm: Hỗ trợ tìm kiếm sách nhanh theo từ khóa (Tên sách, Tên tác giả) phục vụ tra cứu.

**Quản lý các danh mục liên quan:** Bao gồm Tác giả (Author), Nhà xuất bản (Publisher) và Thể loại sách (Category). Các chức năng này đảm bảo tính nhất quán và chuẩn hóa dữ liệu khi nhập sách mới.

### c. Nhóm chức năng Nghiệp vụ Bán hàng

Đây là phân hệ cốt lõi phục vụ hoạt động kinh doanh hàng ngày:

**Quản lý Khách hàng:** Lưu trữ và quản lý thông tin khách hàng (Họ tên, SĐT, Địa chỉ).

#### **Lập hóa đơn:**

- Tìm kiếm thông tin khách hàng qua tên hoặc số điện thoại. Hỗ trợ thêm nhanh khách hàng mới ngay tại giao diện bán hàng để tiết kiệm thời gian giao dịch.
- Giao diện bán hàng trực quan cho phép chọn sách, nhập số lượng mua.
- Kiểm tra tồn kho tự động: Hệ thống tự động so sánh số lượng mua với số lượng tồn kho hiện có. Nếu không đủ hàng, hệ thống sẽ đưa ra cảnh báo và ngăn chặn giao dịch để tránh bán không.
- Tính toán tự động: Tự động tính tổng tiền cho từng mặt hàng và tổng giá trị hóa đơn.

**Xử lý giao dịch:** Sử dụng kỹ thuật Database Transaction để đảm bảo tính toàn vẹn dữ liệu: Khi hóa đơn được lưu thành công, số lượng tồn kho của các cuốn sách tương ứng sẽ tự động được trừ đi. Nếu có bất kỳ lỗi nào xảy ra trong quá trình này, toàn bộ giao dịch sẽ được hủy bỏ để đảm bảo dữ liệu không bị sai lệch.

**In hóa đơn:** Hỗ trợ xuất hóa đơn ra file hoặc in trực tiếp cho khách hàng.

### d. Nhóm chức năng Báo cáo và Thống kê

Cung cấp cái nhìn tổng quan về hiệu quả kinh doanh (Dành cho cấp Quản lý - Admin):

- Báo cáo doanh thu: Thống kê tổng doanh thu và số lượng đơn hàng bán ra trong một khoảng thời gian tùy chọn (Từ ngày... Đến ngày...).
- Thống kê sách bán chạy (Top Selling): Liệt kê danh sách các cuốn sách có doanh số cao nhất, giúp người quản lý nắm bắt xu hướng thị trường.
- Phân tích theo danh mục: Báo cáo chi tiết doanh thu theo từng đầu sách kèm theo thông tin thể loại, giúp đánh giá hiệu quả.
- Xuất dữ liệu: Hỗ trợ xuất các báo cáo thống kê ra định dạng Excel và PDF để lưu trữ, in ấn hoặc phục vụ công tác kế toán.

## CHƯƠNG II. MÔ HÌNH THỰC THỂ

### 1. Tổng quan cơ sở dữ liệu

Hệ thống "Quản lý Nhà sách" sử dụng Hệ quản trị cơ sở dữ liệu **Microsoft SQL Server**. Quy trình thiết kế, truy vấn và tương tác dữ liệu được thực hiện thông qua **Entity Framework Core (EF Core)** theo phương pháp **Code First**.

Các bảng (Table) trong cơ sở dữ liệu không được tạo thủ công bằng lệnh SQL mà được định nghĩa thông qua các lớp C# (Class) trong tầng "BookStore.Entities". Cấu trúc Database, bao gồm các bảng, cột, khóa chính, khóa ngoại và dữ liệu mẫu, được tự động sinh ra và đồng bộ hóa thông qua cơ chế Migration của Entity Framework.

### 2. Danh sách các thực thể (Entities)

#### a. Nhóm Quản trị hệ thống & Phân quyền

**Vai trò (bảng Roles):** Quản lý danh sách các vai trò trong hệ thống (Admin, Staff). Mỗi vai trò sẽ quy định mức độ truy cập khác nhau vào các chức năng của phần mềm.

**Người dùng (bảng Users):** Lưu trữ thông tin tài khoản của nhân viên sử dụng phần mềm. Các thuộc tính quan trọng bao gồm: Tên đăng nhập (UserName), Mật khẩu (Password), Tên hiển thị (FullName) và Vai trò (RoleID) để xác định quyền hạn.

#### b. Nhóm Quản lý Danh mục Sách

**Thể loại (bảng BookCategories):** Danh mục phân loại sách (Ví dụ: Sách giáo khoa, Tiểu thuyết, Kinh tế...). Giúp việc quản lý hàng hóa khoa học và hỗ trợ thống kê doanh thu theo nhóm hàng.

**Tác giả (bảng Authors):** Lưu trữ thông tin của tác giả gồm Tên (AuthorName) và Năm sinh (BirthYear). Một tác giả có thể có nhiều đầu sách.

**Nhà xuất bản (bảng Publishers):** Lưu trữ thông tin nhà xuất bản và địa chỉ.

**Sách (Books):** Thực thể quan trọng nhất trong nhóm này, lưu trữ thông tin chi tiết của từng đầu sách như: Tên sách (Title), Giá bán (Price), Số lượng tồn kho (Stock). Thực thể Book đóng vai trò trung tâm và liên kết chặt chẽ đến 3 bảng danh mục trên: Category, Author và Publisher.

#### c. Nhóm Nghiệp vụ Bán hàng

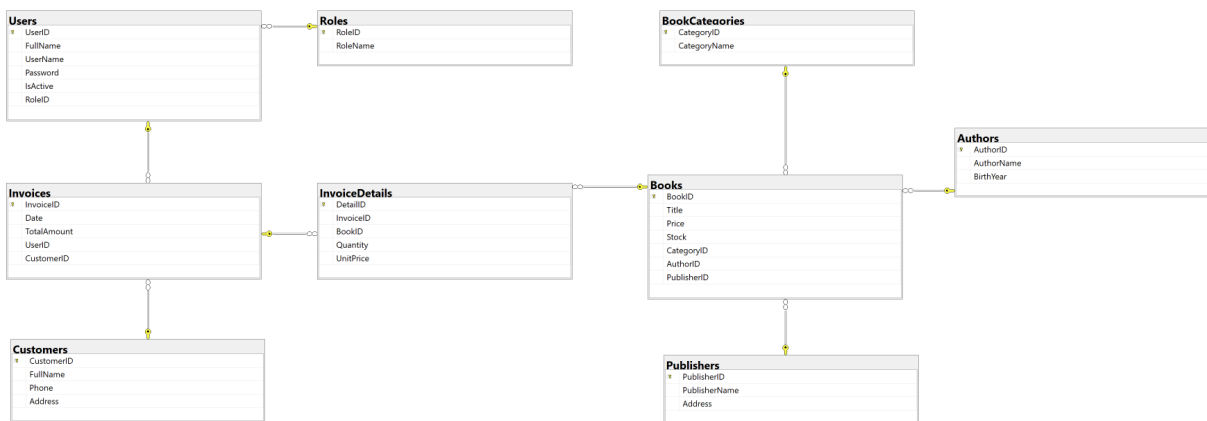
**Khách hàng (bảng Customers):** Lưu trữ hồ sơ khách hàng như: Họ tên (FullName), Số điện thoại (Phone), Địa chỉ (Address) để phục vụ tra cứu và in hóa đơn.

**Hóa đơn (bảng Invoices):** Đại diện cho một giao dịch bán hàng. Lưu trữ các thông tin chung: Ngày lập (Date), Tổng tiền (TotalAmount), Nhân viên lập đơn (UserID) và Khách hàng (CustomerID).

**Hóa đơn chi tiết (bảng InvoiceDetails):** Lưu trữ chi tiết từng mặt hàng trong một hóa đơn. Mỗi dòng trong bảng này ghi nhận: Sách nào được bán (BookID), Số lượng bao nhiêu (Quantity) và Đơn giá tại thời điểm bán (UnitPrice).

#### d. Các lớp hỗ trợ thống kê DTO (Data Transfer Objects)

Các lớp như RevenueReport và BookRevenueDto được xây dựng nhằm lưu trữ dữ liệu tổng hợp (như tổng tiền, số lượng bán) phục vụ riêng cho việc hiển thị báo cáo doanh thu và thống kê sách bán chạy. Các lớp này chỉ chứa kết quả tính toán từ tầng nghiệp vụ (BLL) và không tạo thành bảng vật lý trong Cơ sở dữ liệu để tránh dư thừa dữ liệu.



Hình 1: Sơ đồ quan hệ thực thể

### 3. Cài đặt chi tiết (Implementation)

Việc ánh xạ từ lớp đối tượng (Entity Class) sang bảng dữ liệu (Database Table) được thực hiện kết hợp giữa ‘Data Annotations’ (các thuộc tính [Key], [Required]) và ‘Fluent API’ trong lớp “BookStoreContext”.

#### a. Cấu trúc lớp thực thể (Entity Class)

Ví dụ về cấu trúc lớp “Book” thể hiện các mối quan hệ tham chiếu:

```

namespace BookStore.Entities
{
    [Table("Books")]
    public class Book
    {
    }
}
    
```

```

[Key]
public int BookID { get; set; }
[Required]
[StringLength(200)]
public string Title { get; set; }
public decimal Price { get; set; }
public int Stock { get; set; }
public int CategoryID { get; set; }
[ForeignKey("CategoryID")]
public virtual BookCategory Category { get; set; }
public int AuthorID { get; set; }
[ForeignKey("AuthorID")]
public virtual Author Author { get; set; }
public int PublisherID { get; set; }
[ForeignKey("PublisherID")]
public virtual Publisher Publisher { get; set; }
public virtual ICollection<InvoiceDetail> InvoiceDetails { get; set; }
}
}

```

## b. Cấu hình quan hệ và ràng buộc (Fluent API)

Trong lớp “BookStoreContext”, phương thức ‘OnModelCreating’ được sử dụng để cấu hình chi tiết các mối quan hệ (Relationships), hành vi xóa (Delete Behavior) và định dạng kiểu dữ liệu.

```

protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    base.OnModelCreating(modelBuilder);
    modelBuilder.Entity<Book>().Property(b => b.Price).HasColumnType("decimal(18,2)");
    modelBuilder.Entity<Invoice>().Property(i => i.TotalAmount).HasColumnType("decimal(18,2)");
    modelBuilder.Entity<InvoiceDetail>().Property(d => d.UnitPrice).HasColumnType("decimal(18,2)");

    // 1. Invoice - User (Nhân viên):
    modelBuilder.Entity<Invoice>()
        .HasOne(i => i.User)
        .WithMany()
        .HasForeignKey(i => i.UserID)
        .OnDelete(DeleteBehavior.Restrict);
    // 2. Invoice - Customer (Khách hàng):
    modelBuilder.Entity<Invoice>()
        .HasOne(i => i.Customer)
        .WithMany(c => c.Invoices)
        .HasForeignKey(i => i.CustomerID)
        .OnDelete(DeleteBehavior.Restrict);
    // 3. Invoice - InvoiceDetail:
    modelBuilder.Entity<InvoiceDetail>()

```

```
.HasOne(d => d.Invoice)
.WithMany(i => i.InvoiceDetails)
.HasForeignKey(d => d.InvoiceID)
.OnDelete(DeleteBehavior.Cascade);
// 4. Book - Các bảng cha (Author, Publisher, Category)
modelBuilder.Entity<Book>()
.HasOne(b => b.Author)
.WithMany(a => a.Books)
.HasForeignKey(b => b.AuthorID)
.OnDelete(DeleteBehavior.Restrict);
}
```

Việc cấu hình ‘DeleteBehavior.Restrict’ (Hạn chế xóa) cho các bảng danh mục và ‘DeleteBehavior.Cascade’ (Xóa lan truyền) cho các bảng chi tiết thể hiện sự chặt chẽ trong thiết kế hệ thống, ngăn ngừa các thao tác xóa nhằm gây mất mát dữ liệu quan trọng.

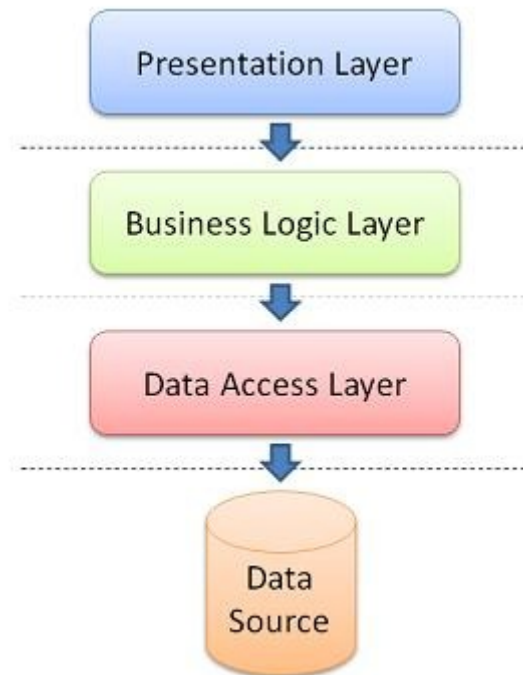
### CHƯƠNG III. THIẾT KẾ CÁC THÀNH PHẦN PHẦN MỀM

#### 1. Kiến trúc tổng thể (Architecture)

Hệ thống quản lý nhà sách được xây dựng dựa trên mô hình **Kiến trúc đa tầng (N-Layer Architecture)**, cụ thể là mô hình 3 lớp kết hợp với **Dependency Injection (DI)**. Việc phân chia này giúp mã nguồn rõ ràng, dễ bảo trì, tách biệt giữa giao diện người dùng, nghiệp vụ xử lý và truy xuất dữ liệu.

Mô hình gồm 4 Components chính:

- **BookStore.UI (Presentation Layer/ GUI):** Tầng giao diện người dùng (Windows Forms).
- **BookStore.BLL (Business Logic Layer):** Tầng xử lý nghiệp vụ, tính toán và giao tiếp giữa UI và DAL.
- **BookStore.DAL (Data Access Layer):** Tầng truy cập dữ liệu, chứa DbContext.
- **BookStore.Entities (DTO):** Chứa các lớp đối tượng ánh xạ với cơ sở dữ liệu.

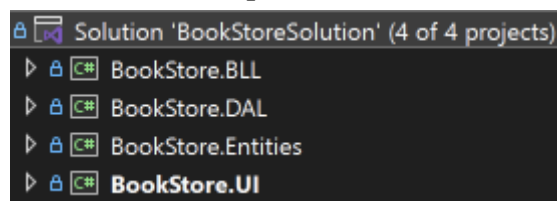


Hình 2: Mô hình kiến trúc 3 lớp của hệ thống

## 2. Tổ chức Dự án (Solution Structure)

Dự án Quản lý nhà sách - Book Store Management được tổ chức thành 4 Components riêng biệt nằm trong cùng một Solution để đảm bảo tính đồng gói:

- **BookStore.UI:** Chứa các Form (MainForm, BookForm...) và file Program.cs cấu hình DI Container.
- **BookStore.BLL:** Chứa các Interface (IBookService) và Class triển khai (BookService) để xử lý logic.
- **BookStore.DAL:** Chứa 'BookStoreContext' kế thừa từ EF Core DbContext.
- **BookStore.Entities:** Chứa các lớp thực thể như Book, User, Invoice...



Hình 3: Cấu trúc Solution

## 3. Mô tả chi tiết quy trình xử lý (Minh họa chức năng)

Để làm rõ cách các tầng tương tác với nhau, phần này sẽ mô tả quy trình thực hiện chức năng phức tạp nhất: "Lập hóa đơn bán hàng" (Create Invoice). Quy trình này đi từ tầng dữ liệu lên giao diện và có sử dụng kỹ thuật Transaction.

## Bước 1: Tầng Entities & DAL (Cơ sở dữ liệu)

Đầu tiên, các thực thể “Invoice” và “InvoiceDetail” được định nghĩa trong BookStore.Entities để lưu trữ thông tin. “BookStoreContext” trong DAL sẽ đóng vai trò cầu nối thao tác trực tiếp với SQL Server.

```
public class Invoice
{
    [Key]
    public int InvoiceID { get; set; }
    public DateTime Date { get; set; } = DateTime.Now;
    public decimal TotalAmount { get; set; }
    public int UserID { get; set; }
    [ForeignKey("UserID")]
    public virtual User User { get; set; }
    public int CustomerID { get; set; }
    [ForeignKey("CustomerID")]
    public virtual Customer Customer { get; set; }
    public virtual ICollection<InvoiceDetail> InvoiceDetails { get; set; }
}
```

*Code minh họa cấu trúc Entity*

```
public class BookStoreContext : DbContext
{
    public BookStoreContext(DbContextOptions<BookStoreContext> options) : base(options) { }
    public DbSet<Role> Roles { get; set; }
    public DbSet<User> Users { get; set; }
    public DbSet<BookCategory> BookCategories { get; set; }
    public DbSet<Author> Authors { get; set; }
    public DbSet<Publisher> Publishers { get; set; }
    public DbSet<Book> Books { get; set; }
    public DbSet<Customer> Customers { get; set; }
    public DbSet<Invoice> Invoices { get; set; }
    public DbSet<InvoiceDetail> InvoiceDetails { get; set; }

    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        base.OnModelCreating(modelBuilder);
        modelBuilder.Entity<Book>().Property(b => b.Price).HasColumnType("decimal(18,2)");
        modelBuilder.Entity<Invoice>().Property(i => i.TotalAmount).HasColumnType("decimal(18,2)");
        modelBuilder.Entity<InvoiceDetail>().Property(d => d.UnitPrice).HasColumnType("decimal(18,2)");

        modelBuilder.Entity<Invoice>()
            .HasOne(i => i.User)
            .WithMany()
            .HasForeignKey(i => i.UserID)
            .OnDelete(DeleteBehavior.Restrict);

        modelBuilder.Entity<Invoice>()
```



```

        .HasOne(i => i.Customer)
        .WithMany(c => c.Invoices)
        .HasForeignKey(i => i.CustomerID)
        .OnDelete(DeleteBehavior.Restrict);

modelBuilder.Entity<InvoiceDetail>()
    .HasOne(d => d.Invoice)
    .WithMany(i => i.InvoiceDetails)
    .HasForeignKey(d => d.InvoiceID)
    .OnDelete(DeleteBehavior.Cascade);
    }
}

```

*Code minh họa cấu hình Context*

## Bước 2: Tầng BLL (Xử lý nghiệp vụ & Transaction)

Tại tầng này, “InvoiceService” thực hiện logic nghiệp vụ quan trọng:

- Khởi tạo Giao dịch (Transaction): Mở một Database Transaction để đảm bảo tính nguyên vẹn dữ liệu.
- Xử lý chi tiết & Kho hàng (Vòng lặp):
  - Duyệt qua từng cuốn sách trong giỏ hàng.
  - Kiểm tra tồn kho (Stock Check): Truy vấn DB xem sách còn đủ số lượng không. Nếu không đủ (Stock < Quantity) thì sẽ ném ra Exception ngay lập tức để dừng quy trình.
  - Trừ tồn kho (Update Stock): Nếu đủ, tiến hành trừ số lượng tồn kho của sách đó.
  - Lưu chi tiết: Ghi nhận dòng bán hàng vào bảng “InvoiceDetails” với giá bán tại thời điểm đó (UnitPrice).
- Cam kết (Commit) hoặc Hoàn tác (Rollback):
  - Nếu tất cả các bước trên thành công thì khởi chạy CommitAsync(): Dữ liệu được lưu vĩnh viễn vào SQL.
  - Nếu có bất kỳ lỗi nào (lỗi mạng, lỗi logic, hết hàng), khởi chạy RollbackAsync(): Hủy toàn bộ các thao tác trước đó, kho hàng và hóa đơn trở về trạng thái ban đầu.

```

public async Task CreateInvoiceAsync(Invoice invoice, List<InvoiceDetail> details)
{
    using (var transaction = await _context.Database.BeginTransactionAsync())
    {
        try
        {
            decimal finalTotal = 0;
            await _context.Invoices.AddAsync(invoice);
            await _context.SaveChangesAsync();

```

```

foreach (var item in details)
{
    item.InvoiceID = invoice.InvoiceID;
    var bookInDb = await _context.Books.FindAsync(item.BookID);
    if (bookInDb == null) throw new Exception("Sách không tồn tại!");
    if (bookInDb.Stock < item.Quantity)
        throw new Exception($"Sách '{bookInDb.Title}' không đủ hàng! (Còn: {bookInDb.Stock})");
    bookInDb.Stock -= item.Quantity;
    item.UnitPrice = bookInDb.Price;
    finalTotal += (item.Quantity * item.UnitPrice);
    await _context.InvoiceDetails.AddAsync(item);
}
invoice.TotalAmount = finalTotal;
await _context.SaveChangesAsync();
await transaction.CommitAsync();
}
catch
{
    await transaction.RollbackAsync();
    throw;
}
}
}

```

*Code xử lý tại InvoiceService*

### Bước 3: Tầng UI (Giao diện người dùng)

Tại “InvoiceForm”, khi người dùng nhấn nút ‘Thanh toán’, hệ thống sẽ gọi xuống BLL thông qua Interface “IInvoiceService” đã được tiêm vào (Inject) qua Constructor.

```

private async void btnSaveInvoice_Click(object sender, EventArgs e)
{
    if (cbCustomer.SelectedValue == null) { MessageBox.Show("Select Customer."); return; }
    if (_cartItems.Count == 0) { MessageBox.Show("Cart is empty."); return; }
    int customerId;
    if (!int.TryParse(cbCustomer.SelectedValue.ToString(), out customerId))
    {
        MessageBox.Show("Invalid Customer."); return;
    }
    try
    {
        btnSaveInvoice.Enabled = false;
        btnAddItem.Enabled = false;
        dgvItems.Enabled = false;
        Cursor = Cursors.WaitCursor;
        var invoice = new Invoice
        {

```

```

        CustomerID = customerId,
        UserID = Program.CurrentUser.UserID,
        Date = DateTime.Now,
        TotalAmount = _totalAmount
    };
    var details = _cartItems.Select(x => new InvoiceDetail
    {
        BookID = x.Book.BookID,
        Quantity = x.Quantity,
        UnitPrice = x.UnitPrice
    }).ToList();
    await _invoiceService.CreateInvoiceAsync(invoice, details);
    MessageBox.Show($"Invoice saved successfully! ID: {invoice.InvoiceID}", "Success");
    _lastSavedInvoiceId = invoice.InvoiceID;
    btnPrintInvoice.Enabled = true;
    _cartItems.Clear();
    UpdateGridAndTotal();
    cbCustomer.SelectedIndex = -1;
    cbBook.SelectedIndex = -1;
}
catch (Exception ex)
{
    MessageBox.Show("Failed to save invoice:\n" + ex.Message, "Transaction Error");
    btnSaveInvoice.Enabled = true;
    btnAddItem.Enabled = true;
    dgvItems.Enabled = true;
}
finally
{
    Cursor = Cursors.Default;
}
}

```

*Đoạn mã gọi xử lý tại InvoiceForm*

## 4. Các kỹ thuật lập trình nâng cao

### a. Dependency Injection (DI)

Dự án không khởi tạo các đối tượng phụ thuộc một cách thủ công (dùng từ khóa ‘new’) mà sử dụng cơ chế Dependency Injection được tích hợp sẵn trong .NET Core.

- Lợi ích: Giúp giảm sự phụ thuộc chặt chẽ (Loose Coupling) giữa các lớp, dễ dàng thay thế, bảo trì code.
- Thực hiện: Việc đăng ký dịch vụ được thực hiện tập trung tại “Program.cs”. Các Form không cần biết cách tạo Service, nó chỉ việc nhận Service thông qua Constructor.

```

var services = new ServiceCollection();
// 1. Database
string connString = ConfigurationManager.ConnectionStrings["BookStoreContext"].ConnectionString;
services.AddDbContext<BookStoreContext>(options => options.UseSqlServer(connString));
// 2. Services (BLL)
services.AddTransient<IAuthService, AuthService>();
services.AddTransient<IUserService, UserService>();
services.AddTransient<IBookService, BookService>();
services.AddTransient<IAuthorService, AuthorService>();
services.AddTransient<IPublisherService, PublisherService>();
services.AddTransient<ICustomerService, CustomerService>();
services.AddTransient<ICategoryService, CategoryService>();
services.AddTransient<IInvoiceService, InvoiceService>();
services.AddTransient<IReportService, ReportService>();
// 3. Forms (UI)
services.AddTransient<LoginForm>();
services.AddTransient<MainForm>();
services.AddTransient<BookForm>();
services.AddTransient<AuthorForm>();
services.AddTransient<PublisherForm>();
services.AddTransient<CustomerForm>();
services.AddTransient<UserForm>();
services.AddTransient<InvoiceForm>();
services.AddTransient<ReportForm>();
services.AddTransient<CategoryForm>();

ServiceProvider = services.BuildServiceProvider();

```

## b. Kỹ thuật Database Transaction

Transaction được áp dụng triệt để cho các nghiệp vụ ghi dữ liệu quan trọng liên quan đến tiền và tồn kho. Điều này đảm bảo dữ liệu không bao giờ bị "bán tín bán nghi" (ví dụ: Hóa đơn đã tạo nhưng Kho chưa trừ, hoặc Kho đã trừ nhưng Hóa đơn chưa lưu).

## c. Truy vấn dữ liệu với LINQ (Language Integrated Query)

Hệ thống sử dụng sức mạnh của LINQ để viết các câu truy vấn phức tạp một cách tường minh, dễ đọc ngay trong mã nguồn C# thay vì viết các chuỗi SQL thuần. Kỹ thuật này được sử dụng trong các chức năng Tìm kiếm (Search) và Báo cáo (Report).

```

return await _context.Invoices
    .Where(i => i.Date >= fromDate && i.Date <= toDate)
    .GroupBy(i => i.Date.Date)
    .Select(g => new RevenueReport
    {

```

```

        Date = g.Key,
        TotalRevenue = g.Sum(i => i.TotalAmount),
        TransactionCount = g.Count()
    })
    .OrderBy(r => r.Date)
    .ToListAsync();

```

#### d. Mô hình bất đồng bộ (Asynchronous Programming)

Toàn bộ các thao tác truy xuất dữ liệu (I/O Bound) đều sử dụng mô hình Async/Await (Task, SaveChangesAsync, ToListAsync).

- Mục đích: Giúp giao diện phần mềm không bị "đơ" (Not Responding) khi thực hiện các tác vụ nặng như tải danh sách hàng nghìn cuốn sách hay lưu hóa đơn, mang lại trải nghiệm người dùng mượt mà.

```

public async Task<List<Book>> GetAllBooksAsync()
{
    return await _context.Books
        .Include(b => b.Category)
        .Include(b => b.Author)
        .Include(b => b.Publisher)
        .OrderByDescending(b => b.BookID)
        .ToListAsync();
}

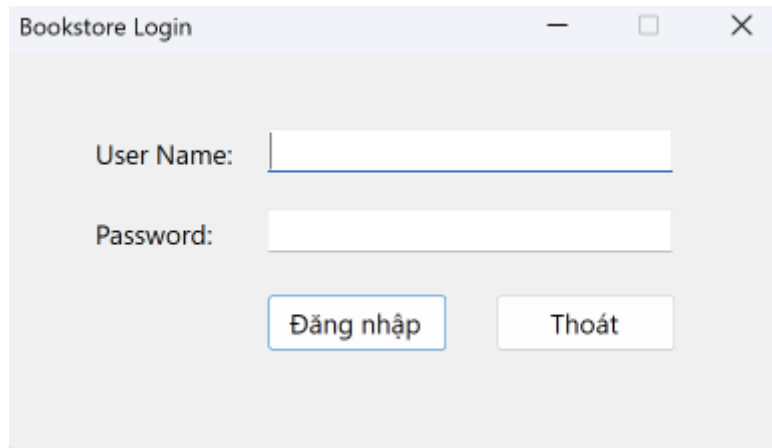
```

## CHƯƠNG IV. CHƯƠNG TRÌNH PHẦN MỀM

### 1. Màn hình Đăng nhập (Login)

Đây là giao diện khởi động của phần mềm, đóng vai trò "cánh cổng" bảo mật. Hệ thống yêu cầu người dùng nhập **Tên đăng nhập** và **Mật khẩu** để xác thực danh tính.

- Cơ chế hoạt động: Khi người dùng nhấn nút "Đăng nhập", hệ thống sẽ gọi xuống tầng Service để kiểm tra thông tin trong Cơ sở dữ liệu.
- Phân quyền: Dựa vào thông tin tài khoản, hệ thống sẽ tự động nhận diện vai trò (Role) là Admin hay Staff để hiển thị các chức năng tương ứng tại màn hình chính.



Bookstore Login

User Name:

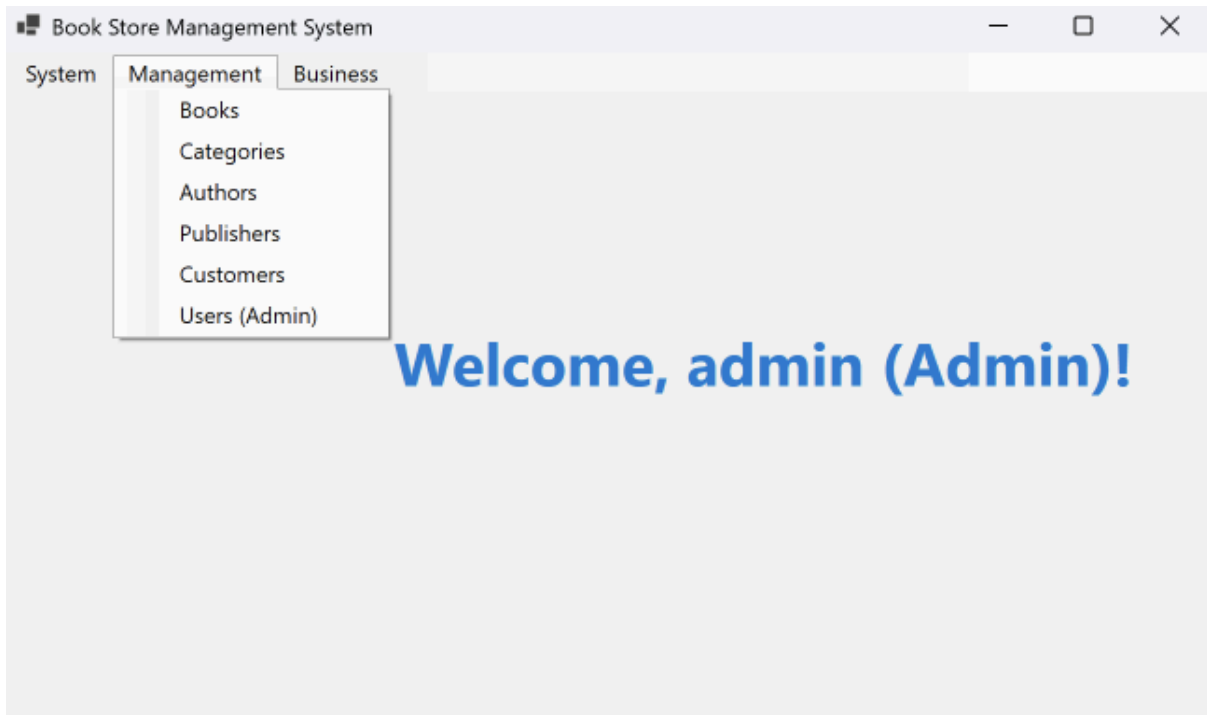
Password:

Hình 4: Giao diện đăng nhập

## 2. Màn hình Chính (Dashboard)

Sau khi đăng nhập thành công, người dùng sẽ được chuyển đến Màn hình chính. Đây là trung tâm điều hướng của toàn bộ ứng dụng.

- Thông tin hiển thị: Hiển thị lời chào mừng, tên nhân viên và vai trò hiện tại (Admin/Staff).
- Thanh điều hướng (Menu): Cung cấp các lối tắt truy cập nhanh vào các chức năng như: Quản lý Sách, Bán hàng, Thống kê, Quản trị hệ thống...
- Cơ chế bảo mật: Nếu đăng nhập với quyền Staff, các menu nhạy cảm như "Quản lý nhân viên" (User) hay "Báo cáo doanh thu" (Report) sẽ tự động bị ẩn đi.



Hình 5: Giao diện màn hình chính

### 3. Phân hệ Quản lý Danh mục (Admin & Staff)

#### a. Quản lý Sách (Book Management)

Đây là chức năng quan trọng nhất để quản lý kho hàng. Giao diện cho phép người dùng thực hiện đầy đủ các thao tác CRUD (Thêm, Xem, Sửa, Xóa).

- Danh sách sách: Hiển thị dưới dạng bảng (DataGridView) với đầy đủ thông tin: Tên sách, Tác giả, Thể loại, Giá bán và Tồn kho.
- Thao tác: Người dùng có thể tìm kiếm sách theo từ khóa, cập nhật giá bán hoặc nhập thêm số lượng tồn kho. Dữ liệu tại đây được liên kết chặt chẽ với các bảng danh mục (Tác giả, Nhà xuất bản) thông qua các ComboBox.

Book Management

Thông tin Sách

Tên sách:

Doraemon Tập 1

Thể loại:

Truyện Tranh (Manga)

Thêm

Giá bán:

25000

Tồn kho:

196

Sửa

Tác giả:

Fujiko F. Fujio

Xuất bản:

NXB Kim Đồng

Xoá

Search by Title...

Tìm kiếm

Xuất file Excel

Xuất file PDF

	Tên sách	Thể loại	Giá bán	Tồn kho	Tác giả	Nhà
	dffdsfb	Truyện Tranh (...)	324,234	14	Fujiko F. Fujio	NXB
▶	Doraemon Tập 1	Truyện Tranh (...)	25,000	196	Fujiko F. Fujio	NXB
	Clean Code (Mã Sạch)	Công Nghệ Th...	450,000	20	Robert C. Martin	O'Re
	Harry Potter và Hòn ...	Tiểu Thuyết & ...	250,000	92	J.K. Rowling	NXB
	Mắt Biếc	Tiểu Thuyết & ...	110,000	43	Nguyễn Nhật Á...	NXB



Thông tin Tác giả

Tên Tác giả:

Năm sinh: ☒ 1/ 1/1933

	Tên Tác Giả	Năm Sinh
▶	Fujiko F. Fujio	1933
	J.K. Rowling	1965
	Nam Cao	1915
	Nguyễn Nhật Ánh	1955
	Robert C. Martin	1952

Hình 7: Giao diện quản lý thông tin Tác giả

Thông tin Thể loại:

Mã:

Tên:

	Mã	Tên Thể Loại
▶	1	Sách Giáo Khoa
	2	Tiểu Thuyết & Văn Học
	3	Kinh Tế & Quản Trị
	4	Công Nghệ Thông Tin
	5	Truyện Tranh (Manga)

Hình 8: Giao diện quản lý thông tin Thể loại sách

Thông tin Nhà xuất bản

Tên:

Địa chỉ:

ID	Tên Nhà Xuất Bản	Địa Chỉ
3	NXB Giáo Dục	81 Trần Hưng Đạo, ...
2	NXB Kim Đồng	55 Quang Trung, H...
1	NXB Trẻ	161B Lý Chính Thăn...
4	O'Reilly Media	USA

Hình 9: Giao diện quản lý thông tin Nhà xuất bản

### c. Quản lý Người dùng (User Management - Admin Only)

Chức năng này chỉ dành riêng cho tài khoản Admin. Quản trị viên có thể:

- Xem danh sách nhân viên trong hệ thống.
- Thêm tài khoản mới và cấp quyền (Role).
- Reset mật khẩu cho nhân viên nếu họ quên mật khẩu.
- Khóa hoặc xóa tài khoản nhân viên đã nghỉ việc.

The screenshot shows a 'User Management' window. At the top, there's a section titled 'Thông tin Khách hàng' (Customer Information). It contains three input fields: 'Tên:' (Name) with the value 'admin', 'Mật khẩu:' (Password), and 'Role:Vai trò:' (Role) with a dropdown menu showing 'Admin'. To the right of these fields are three buttons: 'Thêm' (Add) in green, 'Sửa' (Edit) in blue, and 'Xoá' (Delete) in red. Below this section is a search bar labeled 'Search User...' and a 'Tìm' (Find) button. At the bottom, there is a table with the following data:

ID	Username	Role
1	admin	Admin
2	staff	Staff
8	tetsu	Staff

Hình 10: Giao diện quản lý thông tin Người dùng

#### 4. Phân hệ Bán hàng & Thống kê

##### a. Quản lý Khách hàng (Customer Management)

Giao diện hỗ trợ nhân viên bán hàng tra cứu thông tin khách hàng thân thiết hoặc thêm mới hồ sơ khách hàng ngay tại quầy. Thông tin bao gồm: Họ tên, Số điện thoại và Địa chỉ. Dữ liệu này sẽ được dùng để in lên hóa đơn.

**Customer Management**

Thông tin Khách hàng

Tên:

Số điện thoại:

Địa chỉ:

Search by Name or Phone...

ID	Full Name	Phone	Address
3	Lê Thị B	0918123456	Đà Nẵng
2	Trần Văn A	0909123456	Hà Nội
1	Khách Vãng Lai	0000000000	N/A

Hình 11: Giao diện quản lý thông tin Khách hàng

## b. Lập Hóa đơn bán hàng (Create Invoice)

Đây là chức năng nghiệp vụ phức tạp nhất, nơi diễn ra giao dịch mua bán:

- Quy trình: Nhân viên chọn Khách hàng -> Chọn Sách -> Nhập số lượng -> Nhấn "Thêm" để đưa vào giỏ hàng.
- Tính toán: Hệ thống tự động tính Thành tiền cho từng món và Tổng tiền hóa đơn ('finalTotal').
- Thanh toán: Khi nhấn nút "Thanh toán", hệ thống kích hoạt Transaction để đồng thời lưu hóa đơn và trừ số lượng tồn kho. Nếu kho không đủ, hệ thống sẽ báo lỗi và không cho phép thanh toán.
- In hóa đơn: Sau khi lưu thành công, nhân viên có thể xuất hóa đơn hoặc in.

Lập Hóa Đơn

Thông tin chung

Ngày lập: **17/01/2026 03:37**
Nhân viên:

Khách hàng:

Chọn sản phẩm

Tên sách: 
Số lượng:

	Book Title	Qty	Price	Total	Action
▶	Clean Code (...)	2	450,000	900,000	Remove

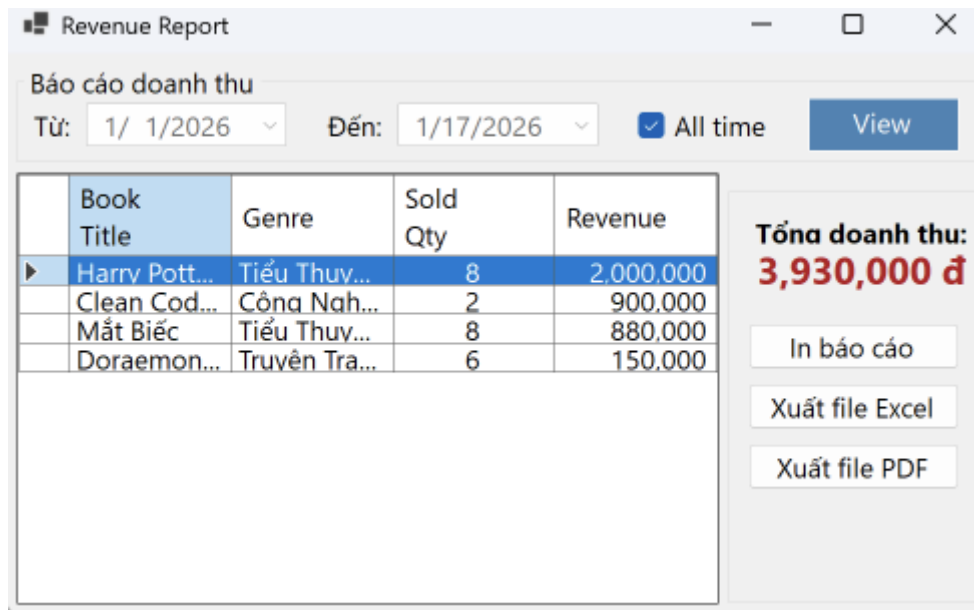
Tổng hoá đơn: **900,000 đ**

Hình 12: Giao diện quản lý thông tin Hoá đơn

### c. Báo cáo Doanh thu (Revenue Report)

Chức năng dành cho quản lý để theo dõi tình hình kinh doanh:

- Bộ lọc: Cho phép chọn khoảng thời gian (Từ ngày - Đến ngày) hoặc xem tất cả.
- Hiển thị: Kết quả hiển thị danh sách các đầu sách đã bán, số lượng bán ra và tổng doanh thu mang lại.
- Xuất báo cáo: Hỗ trợ xuất dữ liệu thống kê ra file Excel hoặc PDF để phục vụ công tác kế toán.



Hình 13: Giao diện quản lý thông tin Thống kê

## CHƯƠNG V. TỔNG KẾT

### 1. Kết luận

Sau quá trình nghiên cứu, phân tích và thực hiện đề tài "Hệ thống Quản lý Nhà sách", em đã hoàn thành các mục tiêu đề ra ban đầu và đạt được những kết quả cụ thể.

#### Về mặt chức năng:

- Xây dựng thành công hệ thống phần mềm đáp ứng đầy đủ các nghiệp vụ cơ bản của một nhà sách: Quản lý danh mục (Sách, Tác giả, Thể loại...), Quản lý khách hàng và nhân viên.
- Hoàn thiện quy trình Bán hàng và Lập hóa đơn, đảm bảo tính chính xác trong việc tính toán tiền và trừ kho tự động.
- Thực hiện thành công chức năng Báo cáo thống kê, giúp người quản lý dễ dàng theo dõi doanh thu theo thời gian và hiệu quả kinh doanh của từng đầu sách.
- Hệ thống có cơ chế Đăng nhập và Phân quyền rõ ràng giữa Quản trị viên (Admin) và Nhân viên (Staff), đảm bảo tính bảo mật dữ liệu.

#### Về mặt kỹ thuật:

- Áp dụng thành công mô hình kiến trúc 3 lớp (3-Layer Architecture) giúp mã nguồn được tổ chức khoa học, rõ ràng và dễ bảo trì.

- Sử dụng thành thạo Entity Framework Core với phương pháp Code First để thiết kế và quản lý cơ sở dữ liệu, giúp việc thay đổi cấu trúc dữ liệu trở nên linh hoạt.
- Triển khai kỹ thuật Dependency Injection (DI) để quản lý các phụ thuộc, giúp giảm sự phụ thuộc chặt chẽ giữa các lớp và tạo điều kiện thuận lợi cho việc kiểm thử, mở rộng.
- Áp dụng kỹ thuật Transaction trong xử lý nghiệp vụ bán hàng, đảm bảo tính toàn vẹn và nhất quán của dữ liệu trong mọi trường hợp.
- Sử dụng mô hình lập trình Bất đồng bộ (Async/Await) giúp ứng dụng hoạt động mượt mà, không bị treo giao diện khi xử lý các tác vụ nặng.

## 2. Các hạn chế

Bên cạnh những kết quả đạt được, phần mềm vẫn còn tồn tại một số hạn chế cần khắc phục:

- Giao diện người dùng: Mặc dù đã đáp ứng được tính năng nhưng giao diện vẫn còn khá đơn giản, chưa thực sự bắt mắt và hiện đại.
- Tính năng mở rộng: Chưa hỗ trợ các tính năng nâng cao như: Tích hợp máy quét mã vạch (Barcode Scanner), gửi hóa đơn qua email, hay quản lý quy trình Nhập hàng (Purchase Order) từ nhà cung cấp.
- Báo cáo: Các báo cáo mới dừng lại ở dạng bảng số liệu, chưa tích hợp các biểu đồ trực quan (Chart) để hỗ trợ phân tích xu hướng tốt hơn.

## 3. Hướng phát triển

Để phần mềm trở nên hoàn thiện và có khả năng ứng dụng thực tế cao hơn, định hướng phát triển trong tương lai bao gồm:

- Nâng cấp giao diện: Sử dụng các bộ thư viện giao diện hiện đại (như GunaUI, Bunifu hoặc chuyển đổi sang WPF) để mang lại trải nghiệm người dùng tốt hơn.
- Tích hợp thiết bị ngoại vi: Bổ sung tính năng hỗ trợ máy quét mã vạch để tăng tốc độ bán hàng và máy in hóa đơn chuyên dụng.
- Mở rộng nghiệp vụ: Phát triển thêm phân hệ Quản lý Kho hàng (Nhập kho, Kiểm kê), Quản lý Công nợ và các chương trình Khuyến mãi/Thẻ thành viên.
- Triển khai đa nền tảng: Nghiên cứu chuyển đổi hệ thống sang nền tảng Web App (ASP.NET Core) hoặc Mobile App để người quản lý có thể theo dõi tình hình kinh doanh từ xa.